2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Attentive Task-Net: Self Supervised Task-Attention Network for Imitation Learning using Video Demonstration

Kartik Ramachandruni, Madhu Babu V, Anima Majumder, Samrat Dutta and Swagat Kumar

***Abstract*— This paper proposes an end-to-end self-supervised feature representation network named *Attentive Task-Net* or AT-Net for video-based task imitation. The proposed AT-Net incorporates a novel multi-level spatial attention module to highlight spatial features corresponding to the intended task demonstrated by the expert. The neural connections in AT-Net ensure the relevant information in the demonstration is amplified and the irrelevant information is suppressed while learning task-specific feature embeddings. This is achieved by a weighted combination of multiple intermediate feature maps of the input image at different stages of the CNN pipeline. The weights of the combination are given by the compatibility scores, predicted by the attention module for respective feature maps. The AT-Net is trained using a metric learning loss which aims to decrease the distance between the feature representations of concurrent frames from multiple view points and increase the distance between temporally consecutive frames. The AT-Net features are then used to formulate a reinforcement learning problem for task imitation. Through experiments on the publicly available Multi-view pouring dataset, it is demonstrated that the output of the attention module highlights the task-specific objects while suppressing the rest of the background. The efficacy of the proposed method is further validated by qualitative and quantitative comparison with a state-of-the-art technique along with intensive ablation studies. The proposed method is implemented to imitate a pouring task where an RL agent is learned with the AT-Net in Gazebo simulator. Our findings show that the AT-Net achieves 6.5% decrease in alignment error along with a reduction in the number of training iterations by almost 155k over the state-of-the-art while satisfactorily imitating the intended task.**

I. INTRODUCTION

Industry 4.0 envisages the autonomous robots working alongside the human worker in an unstructured and dynamic environment. One important requirement of such a collaboration is to have robots with human like learning aptitude, which enables them to learn from interactions with the environment. The idea of such a futuristic setup has been endorsed by the recent advancements in artificial intelligence (AI). In this quest, reinforcement learning (RL) and imitation learning [1], [2], [3] have gained significant attention to make robots more intelligent. Imitation learning, traditionally, has used specific and structured demonstrations in terms of position and velocity in Cartesian / joint space [4], [5], [6]. The demonstrations were generally provided through kinesthetic teaching for a particular skill while manually guiding the robot to perform a task and the motion imitator/model was learned from the demonstrations. Though these methods are computationally efficient and also acquire the expert's skills satisfactorily, they lack the advantages of unsupervised methods and generalization capabilities. Moreover, the human expert's skill is not directly transferred to the robot as the demonstrations involve manual operation of the robotic system.

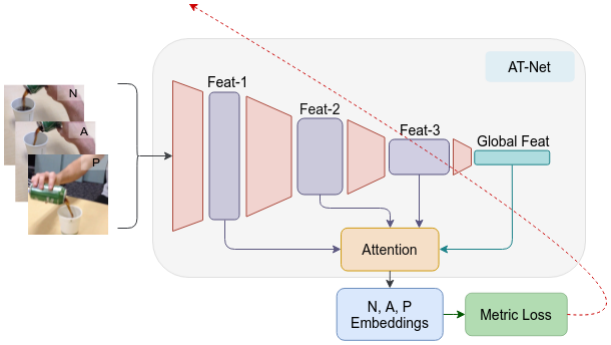


Fig. 1: Overview of the proposed AT-Net. The proposed method has two main modules, namely convolutional feature encoder and multi-level attention module. The convolutional encoder takes three images as input, the positive image, anchor image and negative image and generates task specific feature representations with the aid of the multi-level attention module. These feature representations are then trained using metric loss objective.

In this paper we present an imitation learning framework that helps direct skill transfer from the expert to the robot using unlabeled and unstructured video demonstrations. These demonstrations can be a collection of multiple small tasks[1] involving manipulation, grasping, pulling, pushing etc., thereby making them complex in nature. Thus, to make this skill transfer successful, the learning agent must be able to do the following: a) convert the demonstrations into view point invariant states/abstractions where the states are more tractable, b) understand the important aspects of the demonstrations as image frames may contain information irrelevant to the demonstrated task, c) understand the goal of the task and d) imitate the task. Considering the above attributes, we propose a visual feature representation network that uses multi-view demonstrations to understand the intended task. The proposed network employs a CNN architecture to generate embedding vectors for each frame of a video demonstration. Metric learning is used to bring concurrent frames from different viewpoints together in the embedding space while pulling frames occurring at different time-stamps away from each other. However, this approach does not

The authors are associated with TCS Research, TATA Consultancy Services, Bangalore, India 560066. Email ID: {kartik.vr, madhu.vankadari, anima.majumder, d.samrat}@tcs.com, swagat.kumar@edgehill.ac.uk

[1]An example of such skill transfer can be the mopping task which is a set of small tasks. The expert shows the robot through a video how to use a mop to gather dust and push it into a dust collector.

explicitly ignore irrelevant information that may be present in the video frame such as viewpoint, background, clutter. In order to be truly invariant, the image representation should be focused on the useful parts of the scene which explain the demonstrated task. We achieve this by introducing a multi-level spatial attention module in the proposed feature representation model shown in Fig. 1. The proposed feature extraction module takes care of the first three attributes of the learning agent stated earlier. The feature representations or embeddings obtained from the network can then be used to learn a control policy which imitates the task. The control policy estimation is formulated as a RL problem which exploits the learned embedding vectors. In our case, the RL agent is trained using DDPG algorithm [7]. The reward function of the RL training is designed such that the executed policy drives the robot towards task completion while imitating the expert.

Much work has been done in supervised approaches of imitation learning [8], [9], [10], where state-action trajectories of the expert are recorded either from human demonstrators or expert agents and the algorithms mainly try to replicate these trajectories. In this regard, DQfD algorithm [11] is popular as it was introduced to solve Atari games problem by leveraging demonstration data. There are other methods [12], [13] which takes inspiration from GANs and uses adversarial losses to encourage the agent policy to closely follow the expert policy. To transfer skills through limited visual demonstrations, few-shot methods [14], [15] have also been used. However these methods are unable to learn from abstract demonstrations, such as YouTube videos. Recent works [16], [17] have made significant strides in decoding information from abstract visual demonstrations to tackle this problem. [18] uses visual and audio signals as supervision from a very small sample of YouTube videos of people playing Atari games like PRIVATE EYE and MONTEZUMAS REVENGE. In another work [19], video demonstrations taken from an expert agent are fed to a discriminator along with the imitating agent's video input. The output of the discriminator function is then sent as reward or penalty to the RL agent. Still, these works are currently restricted to the game domain and do not completely address the nuisance variables of real-world video representations such as scale and viewpoint. In this direction, Sermanet et al. [20] uses a triplet loss based feature representation to imitate a pouring task by considering temporal cues from multiple views as the supervision signal without any external supervision. The limitation is that this work considers the entire scene, which makes the learning of the embedding vectors computationally expensive.

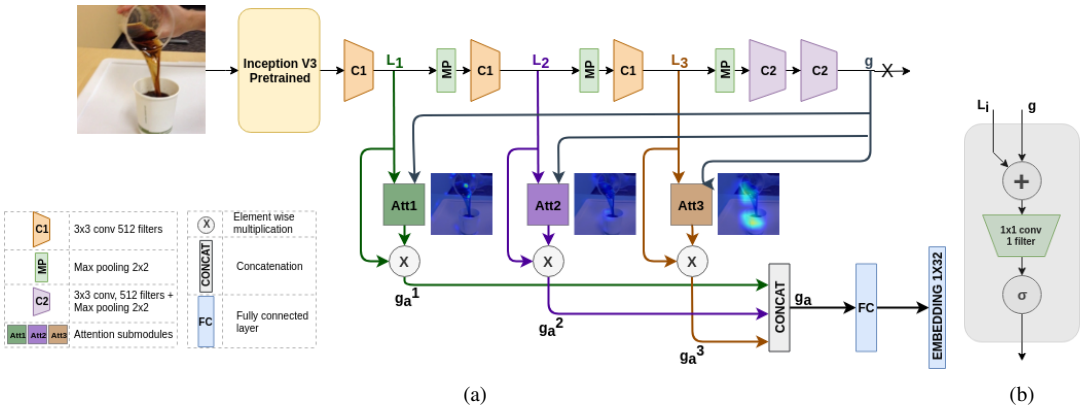
In different contexts, differentiable attention mechanisms have been studied in neural networks for image classification and object recognition [21], [22], [23] as well as action classification and video recognition [24], [25], [26]. Few works also extend the attention concept to learning visuomotor policies [27] and improving deep metric learning [28]. By merging the concept of 'attention' with imitation learning, the RL agent can be made free from the nuisance attributes of the scene. Exploiting this concept, the proposed technique is unique in the way that the multi-level attention module attends to task-specific objects in the spatial domain without any external supervision and extracts this information to generate rich feature representations of the task demonstration. In addition, the proposed network does not require any prior information about the task before learning the feature representation and thus can be generalized to perform on any task. Hence, the contributions made by this work can be summarized as follows:

1) We propose a CNN based feature representation network called ***Attentive-Task Net*** or ***AT-Net*** that incorporates multi-level spatial attention, which is incentivized to amplify the relevant and suppress the irrelevant or misleading information from visual data.
2) The proposed feature representation model is validated using a liquid pouring task where the task imitation is formulated as a reinforcement learning problem. The RL agent uses standard DDPG algorithm to learn the policy.
3) The proposed model reduces the alignment error by almost 6.5% along with a reduction in the number of training iterations by almost 155k over the state-of-the-art. Several ablation studies are also performed on different components of the network in order to validate the proficiency of the generated feature embedding using the proposed model.

The rest of this paper is organized as follows. Section II gives a detailed explanation of the proposed approach and the RL problem statement is defined in Section II-C. Experimental setup and results are presented in Section III in which the Gazebo simulation environment [29] and dataset are introduced in Section III-A. Results obtained upon training are discussed in Section III-B, Section III-C and Section III-D. Finally, in Section IV conclusions are drawn from the work presented.

II. PROPOSED APPROACH

We use time as a supervision signal across multiple viewpoints to perform metric learning. As the time stamps of concurrent frame in the videos are synchronized, the embedding vectors learn what is common among different looking images which are functionally similar, thereby learning features invariant of nuisance variables such as appearance, background and other image related noise. In order to make these embeddings more invariant, we use a multilevel spatial attention module which collects information from feature maps across different depths of the network and highlights the regions among them which are important to the demonstrated task. The spatial attention module helps the network ignore information irrelevant to the task in order to generate an embedding robust towards changing appearances and background. This section first explains the working of the spatial attention mechanism and structure of the overall network followed by a discussion of why this mechanism is successfully in attending to relevant information and ignoring unnecessary details within the image. Thereafter, an RL



(a) (b)

Fig. 2: An architectural overview of the proposed feature representation network. Fig a. shows the proposed network architecture AT-Net, which consists of a CNN pipeline and three spatial attention sub-modules (Att1, Att2 and Att3). These sub-modules calculate the attention weight matrix of each intermediate feature maps (L1, L2 and L3 respectively) with respect to the final layer feature vector g. The weights are then multiplied with the intermediate feature maps to output the attention incorporated feature vector. Fig b. shows the structure of the spatial attention sub-module consisting of an addition operation, a 1x1 convolution operation and a sigmoid activation function.

problem is formulated to solve an imitation learning problem using the proposed architecture.

A. Spatial Attention Structure

The Multi-level Spatial Attention module is shown in Fig. 2a. We use the Inception architecture [30] (initialized with ImageNet pretrained weights) upto the layer 'Mixed_5d' followed by a few extra convolutional layers as the CNN pipeline. Attention sub-modules extract information from multiple layers of this pipeline to generate the required embedding vector. These sub-modules are similar to the ones used in [22], where a multilevel attention mechanism was used for image classification and fine-grained object recognition. Let the set of feature vectors extracted at a given convolutional layer be written as $L^s = \{l_1^s, l_2^s, ...l_n^s\}$, where l_i^s is the vector of output activations at the spatial location $i \in (1,n)$ in the convolutional layer $s \in (1,...,S)$. Also let $\mathbf{g}$ denote the global feature vector, which is essentially the last feature map in the network before the final output layer. In order to incorporate attention into the global feature vector $\mathbf{g}$, we define a compatibility score as follows:

$$C(\hat{L}^s, \mathbf{g}) = \{c_1^s, c_2^s, ...c_n^s\} \tag{1}$$

where $\hat{L}^s$ is the set of vectors of L^s after being linearly mapped to the dimensionality of $\mathbf{g}$. We use the following compatibility score to calculate the relative attention between intermediate feature vector L^s and global feature vector $\mathbf{g}$:

$$c_i^s = < u, l_i^s + g >, i \in \{1...n\} \tag{2}$$

where u is the weight vector learned by a 1x1 convolutional layer that takes the sum of the components as input and gives the compatibility scores as output. The compatibility scores are passed through a sigmoid function to give the normalized compatibility scores $A^s = \{a_1^s, a_2^s, ...a_n^s\}$ as shown in Fig. 2b. These normalized compatibility scores will now function as the attention weights for L^s and are used to produce a single vector by element-wise averaging ($g_a^s = \Sigma_{i=1}^n a_i^s \cdot l_i^s$). The multi-level attention module produces a vector g_a^s for each layer s and these obtained vectors are concatenated to replace the original global vector g with the attention incorporated global vector $g_a = [g_a^1, g_a^2, ...g_a^n]$. This is further passed onto a fully connected layer to produce the final embedding vector.

1) Metric learning: The embedding vectors generated from the network architecture are trained using time-supervised metric learning in order to make them invariant to viewpoint, scaling and other pixel-level changes. We borrow the definitions of anchor, positive and negative examples from the triplet loss literature [31] to explain the training strategy used in our case. The anchor and positive images are defined as concurrent frames taken from different viewpoints, and the negative image as a frame taken from a different time-stamp and arbitrary viewpoint, thereby completing the anchor-positive-negative triplet. In general, metric loss aims to bring the embeddings of the anchor-positive pair closer to each other and pull the embeddings of the anchor-negative pair away. This is done to teach the representation network to cluster samples of similar categories together and push them away from samples belonging to different categories. Hence, in our case, metric learning allows us to learn appearance invariant and task specific feature representations from unlabeled multi-viewpoint video data. Instead of using the triplet loss, we choose to use the multi class N-pair loss [32] as it allows comparison to multiple negative examples per anchor-positive pair and is computationally more efficient.

B. Intuition towards using Attention for Representation learning

To generate an embedding vector which can act as a generalized and nuisance-invariant state representation for training any reinforcement learning agent, it is necessary to encode only those pixels from the video demonstration which are relevant to the imitation task. In a standard CNN pipeline, each layer contains a diverse range of image features and as we go deeper into the network, these features tend to possess more contextual information than spatial information [33]. Our multi-level spatial attention module takes advantage of this behavior by allowing image patches from shallow layers

(local feature vectors l_i^s) to directly contribute to the final embedding vector in proportion to its compatibility with the last layer feature map (global feature vector g). This also means that we are incentivizing the shallow layers to focus on learning those features which are contextually relevant to the imitation task so that they will be embedded in the final representation vector. In addition, similar to the case in [22], there is a greater benefit of using layers relatively late in the network as they are 'relatively mature' and more specific to the task. The use of a multi-level module therefore allows us to access the diversity of information available at different spatial resolutions in the pipeline so that we can generate a more comprehensive and detailed representation vector.

C. RL Framework for Task Imitation

In order to validate the efficacy of the proposed architecture over the baseline architecture, we define an imitation learning task to be solved by an RL agent using the embedding vectors obtained from the network. The task is to mimic the pouring action shown in a single expert video demonstration taken from a customized Gazebo simulation environment [29], shown in Fig. 3. The agent solving this task must do so for any video demonstration irrespective of the viewing angle or distance from which the video is captured i.e. the RL agent must be robust towards scale and viewpoint of the expert. We formulate this problem as a standard Markov Decision Process (MDP) defined by the tuple $< S,A,R,T,\gamma >$ [34], which consists of the set of states, set of actions, reward function, transition function and discount factor respectively. The state representation S of the agent is the embedding vector generated using the image from the simulation environment along with the robot's current state (joint angles and velocities). The action output A of the RL agent is the required joint velocities of the robot. Reward R at time t is defined as:

$$R_t = -\left\|E_{env}(t) - E_{expert}(t)\right\| \tag{3}$$

where $E_{env}(t)$ denotes the current image embedding taken from simulation environment and $E_{expert}(t)$ denotes the embedding of the image in expert video demonstration at time t. γ value is taken as 0.99. A model-free agent, DDPG [7] is used to perform the imitation learning task in the simulation environment because it provides a deterministic output and is suitable for continuous action spaces. We also use the Combined Experience Replay (CER) method for storing experiences [35] as it helps in faster convergence by mitigating the negative effects of a large replay buffer. The method involves appending the most recent experience to the sampled mini-batch from the replay buffer when training the RL network. The expert video demonstration used to train the agent is from a third-person view angle seen during training, but the video itself was not used to train the networks.

III. Experiments and Results

To evaluate the proposed network architecture, we compare its performance to that of the Multi-view TCN architecture [20], which is an appropriate baseline. We first compare both the network architectures when trained on the Multi-view Pouring Dataset [20] by using the validation metrics defined in [20], namely the temporal alignment error and labeled classification error. We then test the proposed architecture on the task imitation problem formulated in Section II-C by developing a simulation environment in Gazebo [29] which replicates the pouring experiment performed in [20]. The performance of the RL agent is compared when using either network to generate state representations. This section presents the results obtained from both comparisons and discusses them in detail.

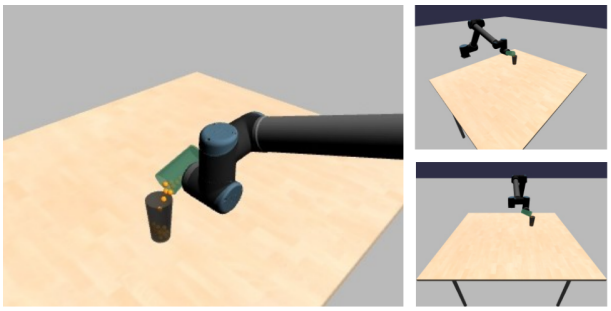


Fig. 3: Images from the Gazebo [29] environment used to replicate the pouring experiment. Beads are being poured from a cup attached to the end-effector of a 6-DOF articulated arm to another cup placed on the table. Videos are captured simultaneously from the first-person view angle (left side image) which is not changed throughout the dataset and the third-person view angle (right side images). Third person videos are taken from a fixed set of yaw angles (viewpoint) and camera distances (scaling) to introduce invariance in the data. The same environment is also used to train the DDPG [7] agent.

A. Network training: Dataset and Validation metrics

The performance analysis of the proposed AT-Net architecture is initially carried out on the publicly available Multi-View Pouring dataset which is a collection of 235 multi-viewpoint video demonstrations of a person pouring different liquids from one container to another. We use two error metrics defined by [20] namely, the temporal alignment error and labeled classification error, which measure the semantic alignment of concurrent images across different viewpoints in the embedding vector space. We also prepared a dataset collected from a simple simulation environment that we developed in Gazebo replicating the pouring experiment performed in [20]. The environment is shown in Fig. 3, where a 6-DOF robotic arm with an attached cup is pouring beads into another cup. The two-view video dataset consists of multiple pouring trajectories collected using random robot and cup positions, thereby ensuring domain randomization in the data. For each pouring demonstration, two cameras are simultaneously recording the trajectory; one fixed camera is from behind the robot (first-person view) and the other is from a different angle facing the robot and table (third-person view). The third-person video demonstrations are made invariant in scale and viewpoint by choosing a fixed set of camera distance and yaw angle values. The collected dataset consists of 212 expert pouring demonstrations in which 175 videos are used for training, 12 videos are used for validation and the rest are left for testing. The validation video set is

Network Architecture	Training iterations	Alignment error	Classification error
Multi-view TCN [20]	224k	17.5%	19.3%
AT-Net Att12	69k	14.1%	16.71%
AT-Net Att23	69k	13.0%	16.01%
AT-Net Att13	69k	12.2%	**15.98%**
AT-Net Att123	**69k**	**10.9%**	16.1%

TABLE I: Comparison between proposed AT-Net architecture and state-of-the-art TCN architecture [20]. Two validation metrics defined by the latter are used to compare the two methods and a significant improvement is seen in the proposed network. Att12 indicates that only the L1 and L2 attention maps are used to generate the embedding vector. Att23 and Att13 similarly follow this notation and Att123 indicates that all three attention maps are being used.

Network Architecture	Training iterations	Alignment error	
		Seen	Unseen
TCN (baseline architecture) [20]	190.5k	7.86%	8.52%
AT-Net (proposed)	**63k**	**4.30%**	**4.07%**

TABLE II: Results obtained on training upon the custom Gazebo [29] pouring dataset. The temporal alignment error is calculated for camera angles seen during training (Seen) and new camera angles (Unseen). The proposed architecture AT-Net performs better than the baseline architecture for both the Seen and Unseen camera view angles.

made such that there exists at least one example from every possible combination of camera distances and yaw angles to have an accurate validation accuracy estimate. An additional set of 75 videos is added to the training set consisting of failed demonstrations such as incorrect pouring or toppling of cup during demonstration. This set of failed demonstrations is necessary as it provides the embedding network with a complete range of possible events that may occur in the environment [20]. We use the temporal alignment error as a validation metric to determine accuracy of the network and perform early stopping for both the proposed network and TCN baseline architecture. Also, a few third-person video demonstrations are taken from outside the fixed set of camera configurations to evaluate the performance of the network for unseen video angles.

B. Quantitative results

The quantitative results of our proposed method are tabulated in Table I and Table II and compared with the TCN baseline architecture. These results are obtained by training on the open-source Multi-view Pouring dataset and custom Gazebo pouring dataset respectively. It is clear from Table I that the proposed method is able to outperform the baseline method with an improvement of 6.5% in the alignment error metric and 3.2% in the classification error metric. In addition, the total number of iterations required for training is reduced by 155k when compared with the baseline method. The baseline requires 224k iterations to converge while our method starts to diverge after 69k iterations. The same trend can be seen in Table II where the proposed method shows an improvement of 3.5% in the alignment error metric for seen view angles and an improvement of 4.5% for unseen view angles. This proves that the attention module is able to filter out the task-irrelevant information and aid the network to reach faster convergence. In addition, the results display the robustness of the method toward variations in the environment which were not shown during training such as new view angles or camera distances.

Network Architecture	No. of parameters	Alignment error	Classification error
AT-Net	1082k	10.92%	16.1%
AT-Net (extended)	1580k	11.9%	17.01%
AT-Net (softmax)	1082k	13.6%	16.64%
AT-Net (Resnet)	1080k	13.1%	18.86%

TABLE III: Ablation study of the network using different network configurations. AT-Net denotes the proposed architecture. AT-Net (extended) is similar to the proposed architecture but with more convolutional layers. AT-Net (softmax) replaces the sigmoid normalization with a softmax normalization. AT-Net (Resnet) is an architecture similar in structure and number of parameters to the AT-Net and is used with a resnet pretrained network.

C. Qualitative results

The attention maps (normalized compatibility scores) generated by our network from some of the images from the test data of the Multi-view Pouring dataset are depicted in Fig. 4. These are visualized to understand where the network is paying attention to while generating the embedding vectors. We observe that the first layer attention weights (L1 attention map) mainly highlight the edges of task relevant objects present within the image such as the hand, liquid and cup. The second layer attention weights (L2 attention map) however focus on highlighting both the cups which are involved in the pouring task. The attention weights from the final layer (L3 attention map) highlight the central region in which the actual pouring is taking place (or where pouring will take place, in case of the images in row 3). In all the attention maps, it is clearly visible that even though different background objects are present they are ignored completely and only the relevant regions of the image with respective to the task are being attended to. The same holds true for the attention maps generated from the Gazebo pouring dataset images (last three rows of images). In addition, although the attention maps of the last layer (last column in the Fig. 4) seem completely inactive in the simulated dataset, the network still outperforms the baseline architecture. This is because unlike real-world images, images from the simulated environment do not contain much distortions or noise, and therefore only two attention layers are required to extract a rich feature representation from them. Furthermore, the attention maps are able to highlight the last image whose view angle was not seen during training. This supports the fact that spatial attention improves domain diversity and allows us to generalize to unseen variations in the environment [22], as is also reflected in the quantitative results.

a) Ablation Study: We carefully designed this network architecture by thoroughly experimenting with the design such as using different combinations of layers for attention, adding extra convolutional layers, changing the activation function of attention layers and using a different pretrained network for extracting initial feature maps. The results are shown in Table I and III respectively. The analysis justifies

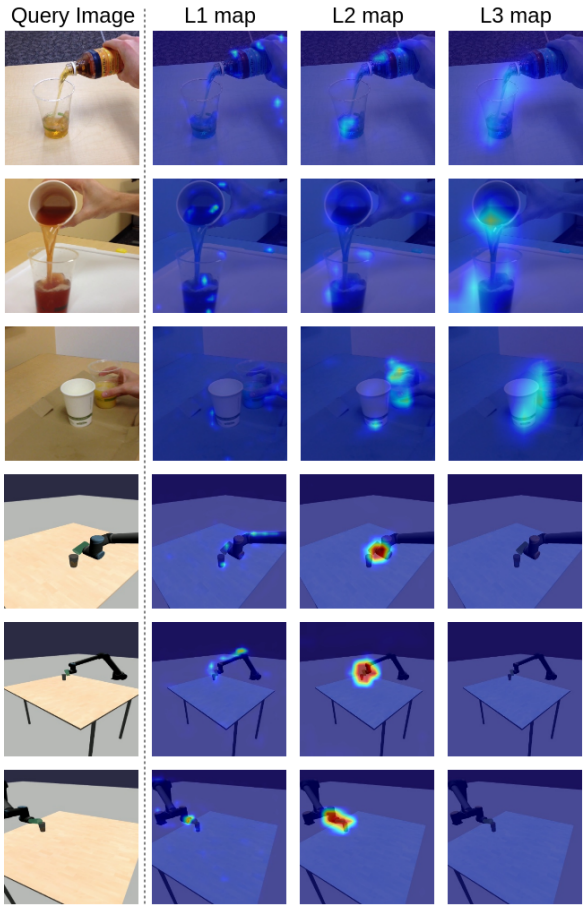


Fig. 4: Attention weight maps generated by the multi-level spatial attention module for multiple images. The last three columns correspond to the three attention maps generated by the multi-level attention module. Top 3 rows of images are taken from the Multi-view Pouring Dataset [20] while the bottom three rows are from the custom Gazebo [29] pouring dataset. The fourth and fifth row images are from first-person viewpoint and third-person viewpoint respectively. The sixth row image is taken from a different view angle not seen during training.

our choice of this network configuration as the proposed architecture.

D. Task Imitation from Video Demonstration

The DDPG [7] agent defined in Section II-C is trained using both TCN and AT-Net architectures to compare their performances. The agent is trained for 4000 episodes each after which the accumulated reward converges and the task is successfully imitated. Fig. 5 shows the accumulated reward of the DDPG agent with time when using both the architectures. The graph indicates that using AT-Net framework allows the RL agent to achieve better rewards (less negative rewards) than when using TCN baseline architecture, even though both the agents are trained using the same set of parameters and same expert video demonstration. This further strengthens the fact that our proposed network is able to align the task representations in the embedding vector space more meaningfully so that they can be used as state representations for a RL or other imitation learning agent.

IV. CONCLUSION

The work proposes a self-supervised representation learning network called AT-Net that uses multi-viewpoint video

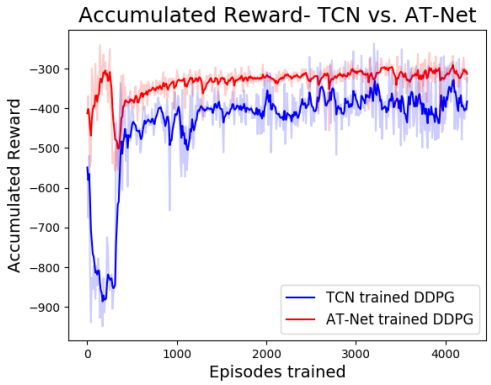


Fig. 5: Accumulated reward by DDPG [7] agent over time when trained using both TCN baseline architecture [20] and proposed AT-Net architecture. The bold line represents a moving average of the actual accumulated reward which is blurred. Both the RL agents are trained using the same hyperparameters and the same expert video demonstration was used. AT-Net trained agent achieves better accumulated reward than the TCN trained agent.

demonstrations to generate a task specific embedding vector for each frame in the demonstrated video. The embedding vector is generated using a multi-level spatial attention framework which captures information from different regions of the input image via a CNN pipeline and highlights the regions relevant to the task shown in the demonstrations. This embedding is further trained using time as a supervision signal across multiple viewpoints using metric learning. The AT-net features along with joint positions and velocities of the robot are then used to represent the states of the RL agent. We formalize the attributes of the imitation learning framework to make successful skill transfer from an expert to the learning agent through video demonstrations. The proposed architecture poses the stated attributes which endow the learning agent to imitate the desired task. We compare the performance of our network with that of the state-of-the-art network TCN [20]. The results obtained demonstrate that spatial attention deployed at multiple levels can improve domain diversity and allow the network to generalize faster, as is observed in the reduction of training steps. We also show that the generated attention maps clearly highlight the objects performing the pouring task while suppressing the background. Further, we perform several ablation studies to understand the impact of implementing spatial attention across multiple levels and justify other network choices. The future scope of this work includes improving the feature representation of our network by providing video frames from sequential time-stamps in order to exploit temporal information such as velocity and acceleration. We will also work towards performing imitation learning with these feature representations by training an RL agent for robotic object manipulation using raw video demonstrations.

REFERENCES

[1] S. Schaal, "Learning from demonstration," in *Advances in neural information processing systems*, pp. 1040–1046, 1997.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[3] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[4] S. Dutta, S. Kumar, and L. Behera, "Learning stable movement primitives by finding a suitable fuzzy lyapunov function from kinesthetic demonstrations," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[5] S. Dutta, L. Behera, and S. Nahavandi, "Skill learning from human demonstrations using dynamical regressive models for multitask applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[6] M. Laskey, C. Chuck, J. Lee, J. Mahler, S. Krishnan, K. Jamieson, A. Dragan, and K. Goldberg, "Comparing human-centric and robot-centric sampling for robot deep learning from demonstrations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 358–365, IEEE, 2017.

[7] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," 2014.

[8] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

[9] W. Sun, A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell, "Deeply aggrevated: Differentiable imitation learning for sequential prediction," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3309–3318, JMLR. org, 2017.

[10] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, IEEE, 2018.

[11] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

[13] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills," *arXiv preprint arXiv:1802.09564*, 2018.

[14] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," in *Advances in neural information processing systems*, pp. 1087–1098, 2017.

[15] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," in *Conference on Robot Learning*, pp. 357–368, 2017.

[16] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125, IEEE, 2018.

[17] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2146–2153, IEEE, 2017.

[18] Y. Aytar, T. Pfaff, D. Budden, T. Paine, Z. Wang, and N. de Freitas, "Playing hard exploration games by watching youtube," in *Advances in Neural Information Processing Systems*, pp. 2930–2941, 2018.

[19] F. Torabi, G. Warnell, and P. Stone, "Imitation learning from video by leveraging proprioception," *arXiv preprint arXiv:1905.09335*, 2019.

[20] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141, IEEE, 2018.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, pp. 2017–2025, 2015.

[22] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv preprint arXiv:1804.02391*, 2018.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[24] D. Li, T. Yao, L.-Y. Duan, T. Mei, and Y. Rui, "Unified spatio-temporal attention networks for action recognition in videos," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 416–428, 2018.

[25] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 305–321, 2018.

[26] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7834–7843, 2018.

[27] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni, "Pay attention!-robustifying a deep visuomotor policy through task-focused visual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4254–4262, 2019.

[28] X. Wang, Y. Hua, E. Kodirov, G. Hu, and N. M. Robertson, "Deep metric learning by online soft mining and class-aware attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5361–5368, 2019.

[29] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2149–2154, IEEE, 2004.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

[32] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016.

[33] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[34] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 2. MIT press Cambridge, 1998.

[35] S. Zhang and R. S. Sutton, "A deeper look at experience replay," *ArXiv*, vol. abs/1712.01275, 2017.